计算机科学
COMPUTER SCIENCE

基于智能映射推荐的知识图谱实例构建与演化方法

张雅晴, 单中原, 赵俊峰, 王亚沙

引用本文

张雅晴, 单中原, 赵俊峰, 王亚沙.基于智能映射推荐的知识图谱实例构建与演化方法[J]. 计算机科学, 2023, 50(6): 142-150.

ZHANG Yaqing, SHAN Zhongyuan, ZHAO Junfeng, WANG Yasha. Intelligent Mapping Recommendation-based Knowledge Graph Instance Construction and Evolution Method [J]. Computer Science, 2023, 50(6): 142-150.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.230300071

基于智能映射推荐的知识图谱实例构建与演化方法

张雅晴[1,2]　单中原[1,2]　赵俊峰[1,2,3]　王亚沙[1,2,3]
1 北京大学计算机学院　北京 100871
2 高可信软件技术教育部重点实验室　北京 100871
3 北京大学(天津滨海)新一代信息技术研究院　天津 300450
(yaqing_zhang@stu.pku.edu.cn)

摘　要　随着大数据技术的深入发展,各领域产生了海量异构数据,构建知识图谱是实现异构数据语义互通的重要手段。通过将结构化数据与本体模型映射匹配来生成实例模型是图谱实例层构建常用的方法。然而,对于复杂异构的领域数据来说,现有映射式实例构建方法大多需要用户手动完成全部映射匹配,映射操作繁琐,无法进行智能匹配,费时费力且容易出错。除此之外,现有方法对实例导入后的增量更新也支持不足。针对现有模式匹配和实例构建方法的映射操作繁琐的问题,提出了基于智能映射推荐的实例构建与演化方法。其中,智能映射复用推荐机制,在用户手动映射之前进行数据模式匹配计算,对元素级相似度、表级相似度和表间传播相似度进行多级相似度综合计算,根据数据模式匹配度仲裁排序后生成推荐映射。另外,增量发现机制通过自动发现冗余实例和冲突实例,生成系统后台任务进行处理,可实现实例的高效无重复导入。在山东市政府开放数据集和深圳市医疗急救数据集上进行了实验,在映射复用推荐模块的辅助下,交互时间缩短为传统模式的约26%,字段推荐匹配准确率达到98.1%;在增量发现模块的实验中,导入了1394万个实例节点以及2158万条关系边所需的时间由31.21h缩短至2.23h,验证了智能映射复用推荐的可用性和匹配准确率,提高了实例层构建与演化的效率。

关键词:知识图谱;模式匹配;映射复用;实例构建;图谱演化
中图法分类号　TP311

Intelligent Mapping Recommendation-based Knowledge Graph Instance Construction and Evolution Method

ZHANG Yaqing[1,2],SHAN Zhongyuan[1,2],ZHAO Junfeng[1,2,3] and WANG Yasha[1,2,3]
1 School of Computer Science,Peking University,Beijing 100871,China
2 Key Laboratory of High Confidence Software Technologies,Ministry of Education,Beijing 100871,China
3 Peking University Information Technology Institute(Tianjin Binhai),Tianjin 300450,China

Abstract　With the development of big data technology,a large amount of heterogeneous data has been generated in various fields. Constructing knowledge graph is an important means to realize semantic intercommunication of heterogeneous data. It is a common method to generate instance model by matching structured data with ontology model mapping. However,most of the existing construction methods require users to manually complete all mapping matching,and the mapping operation is time-consuming and error-prone,unable to perform intelligent matching. In addition,the existing methods do not support incremental updates of the instances. This paper analyzes the existing instance construction methods,and proposes an instance construction and evolution method based on intelligent mapping recommendation to solve the problem of cumbersome manual mapping. Before manually mapping by users,the mapping reuse recommendation mechanism performs multilevel similarity calculation,including element-level similarity,table-level similarity and inter-table propagation similarity,and generates recommendation mapping according to the sorting result of matching. In addition,the incremental discovery mechanism can automatically discover redundant and conflicting instances and generate system background tasks for processing,so as to realize efficient and repeatless import of instances. Experiments are carried out on Shandong government open dataset and Shenzhen medical emergency dataset. With the help of the mapping reuse recommendation module,the interaction time is 3～4 times shorter than that of the traditional mode,and the matching accuracy of field recommendation reaches 98.1%. In the experiment of incremental discovery mechanism,the

到稿日期:2023-03-08　返修日期:2023-04-13
基金项目:国家自然科学基金(62172011);中央高校基本科研业务费
This work was supported by the National Natural Science Foundation of China(62172011) and Fundamental Research Funds for the Central Universities.
通信作者:王亚沙(wangyasha@pku.edu.cn)

time required to import 13.94 million instance nodes and 21.58 million relationship edges is reduced from 31.21h to 2.23h, which proves the availability and matching accuracy of intelligent mapping reuse recommendation, and improves the efficiency of instance layer construction and growth.

Keywords Knowledge graph, Schema matching, Mapping reusing, Instance construction, Graph evolution

1 引言

在大数据时代，各个领域产生的海量异构数据形成了丰富的数据环境，为智慧城市技术及其知识服务提供了巨大的开发空间。如何有效整合海量异构数据，实现异构数据的语义互通和互操作，是智慧城市技术发展所面临的一个重要挑战。数据互操作指多源数据能够实现如同单一系统数据一样的无缝操作。构建大规模知识图谱是实现语义互通和互操作的重要手段。

知识图谱[1]以图模型的形式来建模现实世界中知识和事物之间的联系，在大数据时代对促进知识共享和数据交互具有重大意义。知识图谱的抽象层次可以划分为本体层和实例层两层，其中本体层指本体模型，是对该领域内抽象概念的形式化说明；实例层则是在本体模型的指导下，将各个系统产生的实际数据导入图谱所构成的数据层。由于城市系统产生的数据具有结构复杂、多源异构、高度动态性等特点，在本体模型建立后，如何高效实现实例模型的构建，以及将动态更新的数据源中的实例不断扩充到图谱中，实现知识图谱的动态演化，是大规模知识图谱构建的关键问题。

在城市系统中，大量内部系统产生的数据为结构化数据，其主流存储方式是关系型数据库或 csv/excel 格式的表格。本文针对以结构化数据为数据源进行的知识图谱实例构建，即根据本体模型的概念和关系，建立由数据模式到本体模型的映射，从而完成由结构化数据向知识图谱实例的转化。若要确立上述映射关系，需要在结构化数据的数据模式中找到与本体模型中的概念、属性所对应的表和字段。具体来说，数据模式的表对应本体模型中的概念，数据模式中的字段对应本体中的属性，而其表间连接和外键关系则对应本体模型中的关系。这是一种典型的模式匹配[2]问题。图 1 给出了建立数据模式到本体模型的映射关系的示例。

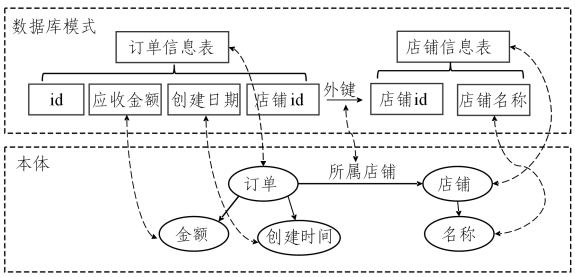

图 1 数据模式到本体模型的映射示例

Fig. 1 Example of mapping from data schema to ontology model

随着模式匹配技术的发展，许多学者提出了一系列自动化模式匹配框架[3]，并在本体映射[4]等领域具有广泛应用。

W3C 的 RDB2RDF 工作组也提出了直接映射[5]（Direct Mapping，DM）和 R2RML[6]两种映射语言，用于将数据库数据转化为 RDF 数据。然而，在领域知识图谱实例构建时，自动化映射方法却存在诸多问题，无法准确高效地实现构建。其主要原因是，在现实的应用场景中，城市系统的表结构往往是由不同业务部门根据自身的业务特点设计的，表结构与本体模型结构差异较大。例如，图 2 给出了某城市医疗系统产生的急救数据，其数据模式包含上百个字段，且存在单张表包含多个概念、概念之间存在上下级的关系。这种情况在现实应用中广泛存在，若进行自动化映射，则需要根据本体模型对数据表进行重构，或者制定复杂的映射规则，这两种方式都十分费时费力。另外，城市系统产生的数据模式通常具有该领域特定的结构，不同来源的数据往往结构差异极大，定制化的映射规则针对异构数据的可泛化性较差，不利于多源异构数据的高效融合。

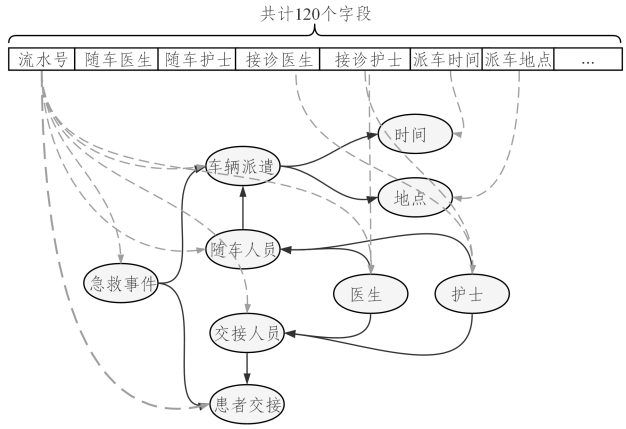

图 2 某城市医疗急救数据模式与本体模型的映射示意图

Fig. 2 Example of complex mapping from schema of urban medical emergency data to its ontology model

为了高效且准确地完成映射构建，现有解决方案大多采用人机交互的形式，由用户指定数据模式到本体模型的映射关系，来指导实例生成。相比自动化映射的方法，人机交互的方法更加灵活，准确率也更高。但现有实例生成工具仍然存在一些问题。目前大多需要用户手动完成全部映射匹配，映射操作繁琐，无法进行智能匹配，费时费力且容易出错。如图 2 所示的例子，若完全进行用户手动映射，则单张表即需要选择上百对映射关系，对用户的耐心是极大的挑战，且容易产生许多错误。另外，用户的映射选择较为依赖专家知识，在没有映射参考或推荐的情况下，对该领域不够了解的用户很难完成准确映射。由于数据模式的动态变化，对于同一个本体模型，其每次映射都需要重新进行整个流程，在数据多源异构的场景下需要消耗大量的时间成本。

随着数据源的不断更新，图谱实例层不断扩充，可能出现实例节点的冗余，人工对实例进行更新十分繁琐。若要实现图谱的动态扩增和自演化，则需要设计实现增量实例的自动发现和查重机制。

针对上述问题，本文提出了一种基于智能映射推荐的

知识图谱实例层构建与演化方法，设计并实现了智慧城市领域知识图谱的实例构建工具；利用智慧城市的医疗和企业领域数据进行实验，验证了本文方法的图谱实例层构建与演化效率。本文方法主要包括实例层构建和实例层演化两个方面：对于实例层的构建，本文提出了智能映射复用推荐机制，在用户手动映射之前进行了数据模式匹配计算，大大简化了用户手动映射的时间成本；对于实例层的演化，增量发现机制通过自动发现冗余实例和冲突实例，生成了系统后台任务进行处理，可实现实例的高效无重复导入，保证了在信息缺失的场景下方法的可用性，并且配合人机交互进行实例融合的确定，支持实例层的自演化。

本文的主要贡献如下：

(1)针对现实城市系统中的实例构建场景，总结了基于结构化数据的知识图谱实例构建的主要方法，并分别从用户交互复杂性、实例构建效率等方面，分析了现有模式匹配和映射式实例构建方法存在的问题。

(2)提出了基于智能映射推荐的知识图谱实例层构建方法，智能映射复用推荐机制在用户手动映射之前进行数据模式匹配计算，对元素级相似度、表级相似度和表间传播相似度进行多级相似度综合计算，根据数据模式匹配度仲裁排序后生成推荐映射。

(3)提出了增量发现机制，用于实例导入阶段自动进行冗余实例节点的查重与合并，支持知识图谱实例层的高效自演化。

(4)在山东市政府开放数据集和深圳市医疗急救数据集上进行了实验，证明了智能映射复用推荐机制和增量发现机制的效率。

2 相关研究工作

2.1 模式匹配与映射

数据模式(schema)是某种特定数据模型的抽象实现，典型的数据模型包括关系数据库模型、XML 数据模型、面向对象数据模型等。模式匹配[7]问题指在两个数据模式的元素之间生成对应关系的问题。

数据模式映射指，对于给定的两个不同的数据模式，通过一定的模式匹配技术，计算得到两个模式元素之间匹配关系的过程。对数据模式与本体模型之间的映射给出形式化定义：对于某个特定的结构化数据模式 S 与本体模型 O，S 与 O 之间的匹配关系 map 由集合表示，如式(1)所示：

$$map:\{m\}=\{\langle u,e,v,rel,f\rangle\} \tag{1}$$

其中，m 表示一个由五元组组成的映射单元，u 表示某个映射单元的唯一标识符，e 与 v 分别为结构化数据模式 S 与本体模型 O 中的元素，且满足 $map(e)=v$；r 用于描述 e 与 v 之间的语义匹配关系；f 则表示该映射关系的置信度，即映射关系的相似度。

在计算映射关系的相似度时，由于单一匹配算法容易引起误差，现有的模式匹配框架通常利用多种模式匹配算法，对输入的元素对进行多个特征维度的相似度计算，例如字符串相似度、结构相似度、统计特征相似度等，然后进行相似度的综合计算，对候选匹配对进行人工判定，最终得到元素映射对集合。

RONTO[8]是早期较为典型的由关系型数据模式向本体映射的模式匹配方法，它综合利用语言特征、语义信息、数据类型相似度进行匹配。它提供了 Protégé 插件，用于 OWL 文件直接生成，并开发了人机交互界面以方便用户查看映射结果。

IOSMA[9]是一种迭代优化的模式匹配算法，它借鉴了传统的模式匹配算法，综合并优化了多种已有的模式匹配算法，在迭代的过程中，利用已匹配的信息来优化传统的模式匹配算法，在不断的迭代中提高传统匹配算法在具有本地化特征的数据上的正确率。本文在 IOSMA 的基础上完成了用户初次映射的推荐。

2.2 映射式实例模型构建

映射式实例模型构建指从关系型数据库或 csv/excel 表格中进行知识抽取，建立数据模式到本体模型的映射，从而完成实例模型的生成。这是一类重要的知识抽取方式。

目前，映射式实例模型构建主要有 3 种方法：1)使用 RDB2RDF 映射语言，直接进行由关系型数据库到 RDF 数据的转化；2)使用 ETL(Extract-Transform-Load)工具进行数据格式转换，完成 OWL 文件的生成；3)人机交互映射，通过一系列人机交互流程，用户手动指定由数据模式到本体模型概念属性的对应关系。

RDB2RDF 于 2012 年发布了 DM 和 R2RML 两种映射语言，其中 DM 是将关系数据库映射为 RDF 数据集的标准，它基于表-概念映射、记录-实例映射等方法，提供了本体映射的基础手段；R2RML 则是用于表示从关系数据库到 RD 数据集自定义映射的语言，可以以基于用户自定义的结构和目标词汇表示原有的关系型数据。也有许多工作基于 RDB2RDF 的标准开发了映射式转化工具，如 Ultrawrap Mapper[10]等。然而，领域知识图谱与世界知识图谱不同，其包含大部分特定领域的专业数据，具有概念关系高度复杂、专业术语多、统计特征本地化等特点，采用 RDB2RDF 的方法通常无法满足自动化映射式实例生成的要求。RDB2RDF 主要包括直接映射和定制规则两种，具体来说，直接映射的 RDB2RDF 方法如 DM 只支持处理建模良好的关系数据库表，且存在范式细节保留、外键依赖、数据模式冗余等问题；而定制规则的 RDB2RDF 方法也只支持处理建模良好的关系数据库表，规则定义复杂，人工定制映射规则工作量较大。

Kettle[11]是使用 ETL 方法进行有结构数据知识抽取的代表，它是一款开源的 ETL 工具，通过手动编写代码，进行由输入到输出的数据格式转化和数据抽取。另外 Kettle 还集成了 Neo4j 图数据库插件，可以实现由 MySQL 数据库、csv 文件直接到 Neo4j 图数据库的转换。但由于 ETL 工具主要针对大规模数据格式的转换及其商务应用，对知识图谱实例构建的适配度不够高，映射设计较为繁琐，需要人工编写大量数据预处理和转化代码。

当前，大部分知识图谱构建系统在开发时通常自行完成实例构建功能的开发，并无统一标准。在进行实例模型构建时，由于不同领域的知识图谱的本体模型设计差异较大，数据来源也多种多样，因此设计者一般主要考虑自身的业务需求和数据特点，而不会刻意与同领域中其他构建工具的模式

保持一致，故实例构建工具通常因系统而异、因领域而异。ArcGIS[12]是美国环境系统研究所(ESRI)研发的在线地理信息知识图谱，它提供简单的映射式图谱构建工具，用户选择平台内数据源，通过实体映射和关系映射构造地理信息知识图谱。gbuilder[13]是北京大学王选所数据实验室开发的知识图谱自动化构建平台，支持结构化数据与本体的可视化映射，可转化为 RDF 三元组进行迁移复用。达观数据开发的渊海知识图谱平台[14]支持有结构数据和本体的多阶段人工映射，用户通过实体映射、实体属性映射、关系映射和关系属性映射 4 个交互阶段，来完成映射关系的构造。

上述系统在人机交互方面大多需要用户手动指定所有映射。在现实场景下，不同部门、不同技术手段所获取的结构化数据，常出现数据字段和本体属性多对多映射、数据表和本体概念/关系多对多映射等复杂映射状况，映射操作繁琐，无法进行智能匹配，费时费力且容易出错。用户的映射选择较为依赖专家知识，在没有映射参考或推荐的情况下，对该领域不够了解的用户很难完成准确映射；由于数据模式的动态变化，对于同一个本体模型，其每次映射都需要重新进行整个流程，在数据多源异构的场景下需要消耗大量时间成本。

3 基于智能映射推荐的实例构建演化方法

本节首先介绍了本文提出的实例构建演化方法的整体思路，然后对框架每个部分的设计思路和细节展开介绍。图 3 为本文方法的整体框架图。

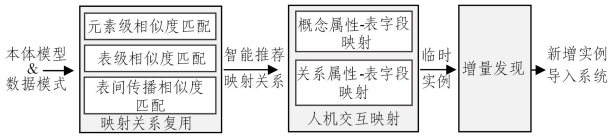

图 3 基于智能映射推荐的实例构建演化方法的整体框架图

Fig. 3 Overall framework of method based on examples recommended by intelligent mapping

3.1 方法概述

为了解决第 1 节中所提到的当前实例构建演化与领域的痛点问题，我们提出了基于人机交互和智能映射推荐的实例构建演化方法。该方法针对结构化数据的实例导入，目标是在本体模型的指导下，结合历史映射数据的复用，生成由数据模式到本体模型的映射关系推荐，经过用户交互映射选择和属性关系歧义消除后，产生待导入系统的新实例节点集合；在生成新实例节点之后，经过增量发现机制来筛选可能出现的冗余节点，待节点导入系统后，调用后台实体对齐任务对新导入实例图谱和原有图谱进行对齐，以完成图谱的高效自增长与自演化。

3.2 智能映射复用与推荐机制

由 2.2 节中对当前映射式实例模型构建现状的分析可知，完全依靠机器进行自动化映射匹配的方法的准确率较低，不适用于领域知识图谱的构建，而大部分人机交互的映射方法要么对数据格式有很多的限制，要么需要人工进行过多的选择与匹配工作，交互流程繁琐且容易出错。如何平衡人机交互工作量和映射生成准确率之间的矛盾，使实例生成过程更加智能，符合人类用户的操作习惯，是设计映射式实例生成方法时需要考虑的最重要的问题。

图 4 给出了一个采购订单模式映射的例子。对于该企业采购信息本体模型 O，在用户上传数据模式 $S2$ 之前，系统中曾经进行过由数据模式 $S1$ 向本体模型 O 的映射。虽然数据模式 $S1$ 与数据模式 $S2$ 的结构和字段不完全相同，但其映射至本体模型 O 的结果却是相同的，如果可以使用 $S1$ 和 O 之间的历史映射，那么理想情况下，我们可以重用所有的映射匹配项，保证 $S2$ 被完全覆盖，不需要额外做任何的匹配工作。

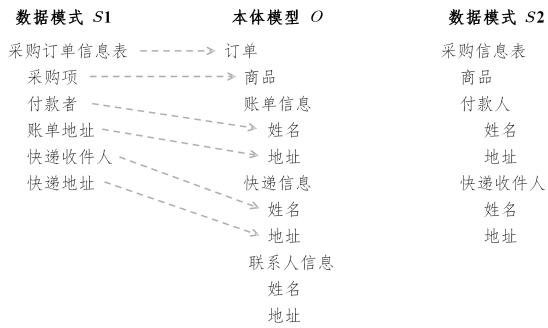

图 4 采购订单模式映射复用示例

Fig. 4 Example of schema mapping reuse for purchase order data

为了减少人工选择映射的工作量和时间开销，本文方法在人机交互映射之前，引入了智能映射复用推荐机制，结合多种模式匹配算法和多版本用户历史映射数据，经过元素级匹配和表级匹配两个层次的相似度计算后，进行相似度综合排序，为用户呈现可复用的映射关系，由用户自行选择是否采用推荐的映射关系。图 5 为映射复用推荐机制的流程框架图。

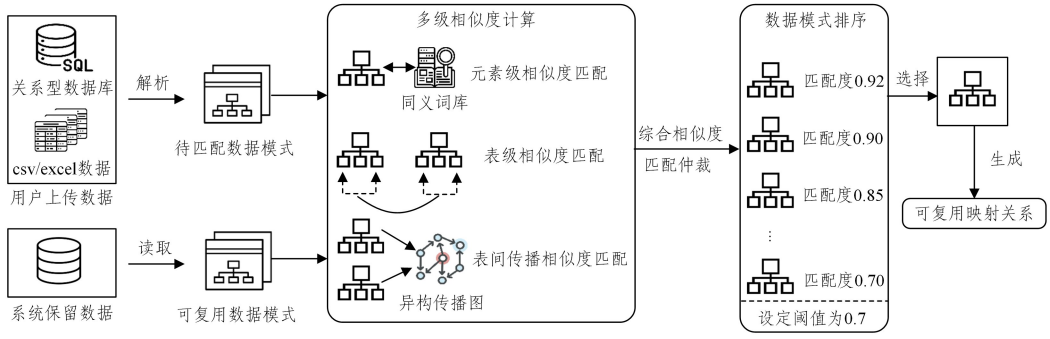

图 5 映射复用推荐机制的流程框架图

Fig. 5 Overall framework of mapping reuse recommendation mechanism

3.2.1 元素级相似度匹配

元素级匹配指在每次映射完成时，将新增映射字段保存在同义词库中；每次用户新建映射时，将待匹配的数据模式的字段与同义词库中的字段结合起来计算字符串相似度，该相似度计算方式如式(2)所示：

$$S_{\text{Thesaurus}}(T,D)=\sum_{t\in T}\min_{1\leqslant i\leqslant s} editSim(t,d_i) \tag{2}$$

其中，t 为待匹配的数据模式的字段，D 为同义词库，d_i 为同义词库中的表项，s 为同义词库的总词条数量。

应当注意，同义词库的建立并不是全局的，应当针对同一概念的关联概念建立同义词关系。若建立全局同义词库，则可能会出现一词多义等歧义。例如，工资和收入在工资单申请中可以被认为是同义词，但在税务报告申请中则并不是相同概念，因此在此例子中，工资和收入这对同义词的建立应当以当前概念为管理单元。

3.2.2 表级相似度匹配

表级匹配的目的是希望重用整个映射结构。我们认为，当待匹配数据模式的结构信息与可复用数据模式的结构越相似，其映射可重用的概率就越大。表级匹配的计算方法是，将待匹配数据模式的每张表与系统中保存的可复用的每张表进行逐表计算，通过计算其表字段相似度、表结构相似度和传播相似度，来得到以单张表为粒度的相似程度。

表字段相似度主要利用表和字段的包含关系来计算两张表的相似度，对其两张表的字段总体相似度进行加权平均，得到整体表字段相似度，其计算式如下：

$$S_{\text{col}}(T,R)=\frac{\sum_{t\in T}\max_{1\leqslant i\leqslant n} editSim(t,r_i)}{colNum_T} \tag{3}$$

其中，$editSim$ 表示字符串编辑距离，T，R 分别代表待匹配的两张表，t 和 r_i 分别为 T，R 中的字段，$colNum_T$ 为 T 的字段数量。

表结构相似度利用表字段的结构信息来计算两张表的相似度，对于关系数据来讲，字段数量是最能体现其二维表结构的信息，其相似度计算式如下：

$$S_{\text{structure}}(T,R)=\frac{\min(colNum_T,colNum_R)}{\max(colNum_T,colNum_R)} \tag{4}$$

其中，T 和 R 分别代表待匹配的两张表，$colNum$ 为表的字段数量。

3.2.3 表间传播相似度匹配

传播相似度用于弥补表结构相似度的不足，进一步考虑了表之间的外键关联关系以及表和字段的包含关系对表相似度计算的影响。根据前人的研究成果[9,15]，本文构建包含“表”和“字段”两种节点的异构传播图，表和表之间的边表示外键关系，表和字段之间的边表示包含关系，则传播相似度可以表示为：

$$S_{\text{propagate}}(T,R)=S_{\text{col2table}}(T,R)*S_{\text{table2table}}(T,R) \tag{5}$$

其中，$S_{\text{col2table}}$ 表示表到字段的传播关系，$S_{\text{table2table}}$ 表示表到表的传播关系。

3.2.4 相似度综合排序与推荐

根据上述计算过程，最终得到的相似度可以综合表示为：

$$S(T,R)=\delta\times S_{\text{Thesaurus}}(T,D)+(1-\delta)\times(S_{\text{col}}(T,R)+S_{\text{structure}}(T,R)+S_{\text{propagate}}(T,R)) \tag{6}$$

其中，超参数 δ 用于平衡元素级相似度和表结构相似度的权重。

由于本系统维护多版本历史映射数据，既包括本用户的历史上传数据模式，也包括其他用户的允许映射复用的数据模式。在进行计算和推荐时，为了保证最终推荐结果的全面性，我们允许最终的映射结果来源于不同的数据模式，即以表为单位进行映射推荐，综合考虑所有数据模式的相似度。如表 1 所列，Ti 表示当前待匹配数据模式所包含的表，schema_k 表示所有的可复用映射版本，表格中出现的元素表示当前表与该数据模式中的表相似度大于所设阈值(本系统设定相似度阈值为 90%)，根据表匹配数量排序，得到最相似的推荐数据模式为 Schema_4。

表 1 按表匹配数目对数据模式排序

Table 1 Sorted data schema by the number of table matches

	0	T1	T2	T3	T4	T5	T6	表匹配数目
Schema_4	R4_5		R4_0	R4_7		R4_1	R4_3	5
Schema_2	R2_2	R2_3		R2_9		R2_4		4
Schema_3			R3_1	R3_9			R3_3	3
Schema_1		R1_0		R1_3				2
Schema_0			R0_5					1
Schema_5								0

3.3 人机交互实例映射

在领域知识图谱的实例生成过程中，由于数据源存在多源异构的复杂情况，机器推荐无法保证绝对正确，因此需要人工进行交互映射来进行选择。本文方法实现了用户交互映射界面，在经过智能映射推荐之后，用户可以选择采纳推荐，之后对该推荐进行检查和修改，或选择不采纳推荐，完全重新进行映射选择。

由 2.2 节的分析可知，在现实场景下，来源于不同系统或部门的数据模式往往设计差异较大，常出现数据字段和本体属性多对多映射、数据表和本体概念/关系多对多映射等复杂映射状况，现有映射式实例生成工具均无法提供很好的支持。本文方法根据现实场景中的复杂业务需求，对于数据模式较为复杂的表，允许同一张表映射到多个概念、同一张表的字段映射到多个概念的多个属性，保证了映射的灵活性；通过“表-概念对应”的结果，来限制表和关系的对应，以保证表和关系的映射结果是合法的。

由于在映射规则上缺少了对表到概念的一对一限制关系，以及关系的头尾节点的一对一限制关系，直接由用户选择的映射关系生成实例可能会出现歧义。因此本文方法在映射选择交互结束后，设计了属性歧义消除和关系歧义消除两个交互流程，用于消除多对多映射可能带来的歧义，确保实例生成的准确性。

进行映射选择和歧义消除之后，收集实例构建所需要的信息，系统会生成临时实例，进入图谱增长和演化阶段。

3.4 增量发现机制

增量实例指由用户上传的数据生成的、实例层中尚未包含的新实例；只有增量实例才可以被导入实例层。与之相反，已有实例指由用户上传的数据生成的、实例层中已经包含的旧实例；已有实例不允许重复导入。若已有实例和实例层中对应的实例存在属性值冲突，则这个已有实例就被称为冲突实例。在实例生成任务结束后，系统为冲突实例生成后台任务，由人来解决属性值冲突。

如果当前是实例层构建场景，则实例库为空，系统默认数据中包含的所有实例及关系均为增量。如果当前是实例层增长场景，则系统会在实例库中，利用存储系统的索引机制，对新数据中的实例进行快速查重，发现增量实例和冲突实例。增量实例导入图谱，冲突实例生成用户任务留待用户解决。系统也会动态地进行关系查重，发现增量关系。通过增量发现阶段，过滤重复数据，提高实例生成的效率。

首先，在增量实例集合中根据临时实例的主属性值进行查重。如果发现增量实例集合中有重复实例，则将临时实例的属性值合并到重复实例中；如果没有发现重复，则说明这个临时实例尚未被其他数据记录生成过，但这并不代表这个临时实例就可以被判定为增量，因为实例库中可能有主属性值相同的重复实例，此时进入下一子步骤。

其次，在库实例映像集合中根据临时实例的主属性值进行查重，库实例映像集合是实例库中被访问过的实例在内存中的缓存，库实例映像集合减少了访问实例库的次数，提高了查重的效率。如果发现库实例映像集合中有重复实例，则说明实例库中已经存在主属性值相同的重复实例，这个临时实例是已有实例；还要对比临时实例的非主属性值和库实例映像集合中对应实例的非主属性值，如果发现属性值冲突，则说明这个临时实例是冲突实例，系统记录其属性值冲突信息，并将其汇总成冲突解决任务，留待用户统一解决。如果没有发现重复，则进入下一步骤。

最后，访问实例库，根据临时实例的主属性值进行查重，实例库为实例的主属性建立了索引，保证等值检索的效率。如果发现实例库中有主属性值相同的重复实例，临时实例即为已有实例，系统将实例库中的对应实例取出并存入库实例映像集合中，之后进行非主属性值对比并生成任务（与上一子步骤相同）。如果实例库中没有发现重复，则说明这个临时实例是增量，将临时实例存入增量实例集合中。

基于先查内存再访问库的机制、库实例映像集合的内存缓存机制、实例库的索引机制，在实例生成阶段，可以高效地发现增量实例。实例生成结束后，增量实例集合中存储了所有的增量实例节点，库实例映像集合中存储了所有的已有实例节点。

4 工具实现与验证

4.1 工具实现

根据本文方法，设计并实现了一个基于智能映射推荐的实例构建演化系统。本系统基于微服务架构，自底向上包括数据存储层、核心算法层、系统业务层和前端展示层 4 个层次。图 6 为系统实现架构图。

数据存储层采用 MySQL 关系数据库、Dgraph 图数据库和 MongoDB 文档数据库的混合存储模式，目的是提高本体和实例的读取导入查询效率；核心算法层包括本文方法提出的几大模块，分别为实例生成、关系生成、映射复用和增量发现模块，在本层完成了逻辑实现；系统业务层基于 Spring Boot[15] 框架，采用 RESTful 风格的 API 设计，实现了系统功能接口用于前端调用，为下层算法实现和数据存储提供了数据支持；前端展示层为用户提供交互友好的系统页面，以本体模型和数据模式列表的形式展示待选择映射，通过点击、拖拽等操作完成映射选择和歧义消除，实现高效快速的实例构建。

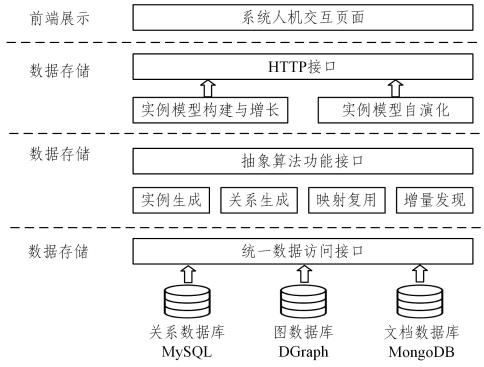

图 6 系统实现总体架构图

Fig. 6 Overall architecture of system implementation

下文截取部分系统前端页面来展示用户可视化交互页面设计。图 7 给出了进行映射选择时用户与系统的交互界面，左侧列表为本体模型，以概念层级树的形式呈现，右侧列表为用户本次上传的数据模式，以表字段层级树的形式呈现。用户通过点击两侧的属性和字段进行映射，完成一次匹配后，就建立了一个“属性-字段”映射对，系统会将本映射对展示在该属性和字段的行尾，支持点击叉号取消映射对。在系统进行映射推荐后，将直接在映射选择界面展示已经匹配完成的映射对，如图 8 所示。

图 7 用户人机交互式映射概念选择界面

Fig. 7 Mapping select of concept page

图 8 用户人机交互式映射属性选择界面

Fig. 8 Mapping select of property page

4.2 实验验证

本节对本文方法进行了实验测试，主要评估指标为推荐准确度和系统可用性。推荐准确度指系统推荐的正确映射对数量占总映射对数量的比例，该比例越高，用户需要的人工手动映射次数越少，系统易用性就越强。系统可用性主要指方法操作是否简单，在用户对计算机和知识图谱系统的了解程度难以量化的情况下，操作越简单，易用性越强。

4.2.1 映射推荐模块实验验证

为了验证本推荐机制针对智慧城市的各个应用场景的数据均具有有效性，本文分别选取了医疗领域、水利工程领域、政府领域、微博舆情领域等领域共计 578 个映射字段数据，均从开放渠道下载或爬取。详细数据字段数如表 2 所列。

表 2 各领域数据表名及其字段数详情

Table 2 Fields number and names of different domain tables

领域	数据表名称	字段数
医疗领域	深圳市 2011—2019 年急救数据表	120
水利工程领域	水利工程信息表	53
	水利气象信息表	60
	水域信息表	32
政府领域	东营市政府开放数据集	51
	德州市政府开放数据集	15
	济宁市政府开放数据集	23
	聊城市政府开放数据集	14
微博舆情领域	用户信息表	98
	评论信息表	43
	帖子相关信息表	69

分别与相应领域的通用本体模型的概念和属性进行匹配，按照元素级、表级和表之间的关系等相似度进行计算，得到最终的综合相似度推荐，正确匹配标签根据专家经验进行定义。经过多次实验，得到每张表的推荐准确率和总的平均准确率，如表 3 所列。

表 3 各领域数据表的字段映射推荐准确率

Table 3 Field mapping recommendation accuracy for each domain table

领域	数据表名称	准确率
医疗领域	深圳市 2011—2019 年急救数据表	0.9834
水利工程领域	水利工程信息表	0.9942
	水利气象信息表	1.0000
	水域信息表	0.9804
政府领域	东营市政府开放数据集	0.9457
	德州市政府开放数据集	0.9648
	济宁市政府开放数据集	0.9645
	聊城市政府开放数据集	0.9583
微博舆情领域	用户信息表	0.9243
	评论信息表	0.9023
	帖子相关信息表	0.9132
平均准确率		**0.9546**

整体推荐准确率在 95%以上，证明了映射推荐模块的可用性，基本可以满足大部分映射场景的需要。除此之外，对于医疗和水利领域这种专业领域的数据而言，由于实验字段具有一定的专业性，例如水利工程中的“警戒水位”“泄洪圩区”等，通过字段相似度即可很快推荐得到。而由于微博领域数据的字段为英文，例如“mid”“resource_url”等，因此仅通过字符串编辑距离较难进行全部匹配，需要通过表结构相似性等信息进行补充，因此整体推荐准确率较低。

4.2.2 映射推荐模块案例研究

案例研究实验通过实际数据集导入来验证该机制是否能够辅助用户进行映射操作，并评估系统实际操作时的推荐准确度。

映射复用模块的验证采用从山东省 4 个地级市的政府开放网站上获取的城市政务数据表作为实验数据集，数据集信息如表 4 所列。

表 4 山东省政府开放数据

Table 4 Information of Shandong open-data collection

城市	数据表名称	字段数
东营	机动车环保检验机构信息	3
	基层法律服务工作者信息	6
	建设用地规划许可证信息	5
	律师执业信息	7
	社会组织注册登记信息	6
	事业单位法人登记信息	10
	行政处罚信息	9
	医保定点药店	3
	医疗保险定点医疗机构信息	2
德州	德州机动车环保检验机构	3
	基层法律服务工作者信息表	7
	建设用地规划许可证	5
济宁	律师执业信息汇总	8
	济宁社会组织注册登记表	6
	事业单位法人信息	9
聊城	行政处罚信息表	9
	医保定点药店	3
	聊城医疗保险定点医疗机构信息	2

为了验证智能映射复用模块对用户交互的辅助效果，我们要求 10 位用户在了解企业信息领域本体的前提下，在不使用映射复用推荐和使用映射复用推荐两种情况下，分别独立地完成映射工作，记录平均点击次数和平均时间。

为了验证映射复用推荐的准确度，我们通过分批迭代式地导入数据集中的数据模式，查看历史映射数据和其他用户映射数据对推荐匹配的效果，以每轮成功推荐的字段数量为衡量指标，在最后一轮上传全部数据时，记录系统成功推荐的字段数量。

在可用性方面，对 10 位用户的操作结果取平均值，如表 5 所列。

表 5 人机交互效率的实验结果

Table 5 Experimental results of user interface efficiency

智能映射复用推荐	平均交互时间/s	平均点击次数/次
使用推荐	481.20	69.40
不使用推荐	1857.40	226.50

可以看到，使用映射复用推荐的平均交互时间为 481.20 s，平均点击次数为 69.4 次，交互时间比不使用推荐的

1857.40 s缩短了近2/3，点击次数比不使用推荐的226.5次减少了近3/4，证明了本系统对人机交互映射式实例生成的有效辅助。

在推荐准确度方面，首先依次上传德州、济宁、聊城的数据，由于这3个城市数据包含的表互相差别较大，系统没有找到可复用的映射关系，均按照字符串编辑距离进行默认推荐。随后上传东营的数据，经过计算后，系统给出了如下的匹配结果：德州、济宁、聊城的数据模式分别匹配到了东营数据中的3张表，这些表对应的映射关系被系统用于生成可复用映射关系，用于辅助用户对东营数据模式和本体进行映射。以“属性-字段”推荐映射对的准确率为验证指标，对于东营市数据表包含的51个字段，系统给出了50个正确的推荐映射对，准确率达到了98.04%，用户只需再对剩下的一个字段操作进行映射即可。

上述实验结果和案例分析验证了本智能推荐匹配算法的有效性。

4.2.3 增量发现模块实验验证

为了证明数据增量发现机制确实能避免重复实例的生成、提高实例生成的效率，通过实际数据集上的实验来验证该机制的有效性。实验数据集采用深圳急救中心两年的120调度数据，数据集信息如表6所列。

表6 深圳急救中心两年的120调度数据统计信息

Table 6 Ambulance dispatching data of Shenzhen emergency center for 2 years

数据表名称	字段数	记录数
2018年120调度数据	108	616791
2019年120调度数据	108	635249

先在不包含增量发现机制的情况下进行实例导入，只导入2018年的120调度数据，数据总共派生出1394万个实例节点以及2158万条实例节点之间的关系。这里对比本文方法不包含增量发现机制的版本和包含增量发现机制的版本，对比的指标是导入实例节点和实例之间关系的总耗时，实验结果如表7所列。

表7 深圳急救中心2018年120调度数据集实验结果

Table 7 Experimental results of KG construction upon Shenzhen emergency ambulance dispatching data

方法	导入总耗时/h
本文方法(包含增量发现机制)	2.23
本文方法(不包含增量发现机制)	31.21

从实验结果可以看出，加入了增量发现机制后，实例及关系的导入效率的提升幅度较大，包含增量发现机制方法的导入总耗时仅为不包含增量发现机制方法的导入总耗时的7%。由此可见，增量发现机制有效地提高了实例及关系的导入效率。

再验证实例生成是否准确，先导入2018年的120调度数据，随后将2018年和2019年的120调度数据合并后再次导入。第二次导入结束后，对于全部有主属性的概念，统计其实例数目。统计结果显示，执行第二次导入后，只有2019年120调度数据中包含的实例被导入实例库中，2018年120调度数据中包含的实例全部被增量发现机制判定为已有实例，不会重复导入。由此可见，增量发现机制能够准确识别增量，避免重复实例的生成。

最后验证冲突实例能否被发现，先导入2018年的120调度数据，随后修改2018年的120调度数据中的某些记录，并再次导入。第二次导入结束后，系统生成冲突解决任务，发现了所有改动的记录对应的实例，并记录其属性值冲突情况。由此可见，增量发现机制能够准确识别冲突实例，保证了实例层增长的准确性。

综上所述，数据增量发现机制能够避免重复实例的生成、提高实例生成的效率，并大大增加实例构建系统的可用性，机制的高效性和准确性得到了验证。

结束语 本文提出了一种基于智能映射推荐的知识图谱实例层构建演化方法，针对现实城市系统中的实例构建场景，总结了现有模式匹配和映射式实例构建方法所面临的主要问题，在保证支持复杂数据模式的条件下，创新性地提出了映射复用机制和增量发现机制，可以高效实现领域知识图谱实例层的构建。结合本文方法，实现了基于智能映射推荐的知识图谱实例构建演化系统，在本系统上进行了实际场景下的导入实验，验证了本文方法的可用性和推荐准确率。

本文对未来的研究工作有如下几点思考和展望。首先，对于映射复用推荐算法的相似度计算，应当探索更加全面和科学的综合计算方案，目前主流的方式都是用字段信息、表结构信息等用户指定的综合方式，然而大多都是基于经验出发，无法完全保证是最佳的综合相似度计算公式。其次，在图谱增长与演化方面，应当尝试除了增量发现以外的其他知识融合方式，目前增量发现的实例比较仍然是针对主属性和非主属性是否完全一致的方式，后续的研究或许可以考虑使用实体对齐等知识融合的方式对新增图谱和原有图谱进行对齐。

参 考 文 献

[1] SINGHAL A. Introducing the knowledgegraph:things,not strings[Z/OL]. Official google blog. 2012 http://googleblog.blogspot.pt/2012/05/introducing-knowledge-graph-things-not.html.

[2] RAHM E, BERNSTEIN P A. A survey of approaches to automatic schema matching[J]. the VLDB Journal, 2001, 10(4): 334-350.

[3] MASSMANN S, RAUNICH S, AUMÜLLER D, et al. Evolution of the COMA match system[J]. Ontology Matching, 2011, 49: 49-60.

[4] SHVAIKO P, EUZENAT J. Ontology matching: state of the art and future challenges[J]. IEEE Transactions on Knowledge and Data Engineering, 2011, 25(1): 158-76.

[5] ARENAS M, BERTAILS A, PRUD'HOMMEAUX E, et al. A

direct mapping of relational data to RDF[J]. W3C recommendation, 2012, 27: 1-11.
[6] SOURIPRIYA DAS S S, RICHARD CYGANIAK. R2RML: RDB to RDF Mapping Language [OL]. https://wwww3org/TR/r2rml/.
[7] BERNSTEIN P A, MADHAVAN J, RAHM E. Generic schema matching, ten years later[C]// Proceedings of the VLDB Endowment. 2011: 695-701.
[8] PAPAPANAGIOTOU P, KATSIOULI P, TSETSOS V, et al. RONTO: Relationalto ontology schema matching[J]. AIS Sigsemis Bulletin, 2006, 3(3/4): 32-36.
[9] WANG F, WANG Y S, ZHAO J F, et al. A Schema Matching Method from relational model to ontology Model Based on Iteration[J]. Journal of Software, 2019, 30(5): 1510-1521.
[10] SEQUEDA J F, MIRANKER D P. Ultrawrap Mapper: A Semi-Automatic Relational Database to RDF(RDB2RDF) Mapping Tool[C]// Proceedings of the ISWC(Posters & Demos). 2015.
[11] Pentaho Data Integration-Pentaho Documentation [OL]. https ://helphitachivantaracom/Documentation/Pentaho/93.
[12] ArcGIS [OL]. https://developersarcgiscom/.
[13] PKUMOD. gBuilder [OL]. http://wwwopenkgcn/tool/gbuilder/.
[14] Spring Boot [OL]. https://springio/projects/spring-boot.
[15] MELNIK S, GARCIA-MOLINA H, RAHM E. Similarity flooding: A versatile graph matching algorithm and its application to schema matching[C]// Proceedings 18th International Conference on Data Engineering. IEEE, 2002: 117-128.

ZHANG Yaqing, born in 1999, postgraduate, is a member of China Computer Federation. Her main research interests is knowledge graph.

WANG Yasha, born in 1975, Ph.D, professor. His main research interests include big data analysis, artificial intelligence, and urban computing.

(责任编辑:喻藜)